IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Multi-Axis Television Navigation

Inventors:
Samuel Thomas Scott, III
David Hendler Sloo
Peter T. Barrett

ATTORNEY'S DOCKET NO. MS1-1731US

COMPUTER PROGRAM LISTING

One .XSD text file listing and one .XML text file listing used in accordance with the subject matter are provided in an appendix after the abstract on 3 sheets of paper and incorporated by reference into the specification. The .XSD text file is an exemplary sample extensible markup language schema definition file for a metadata database and the .XML text file is an exemplary sample database suitable for multi-axis TV navigation.

TECHNICAL FIELD

The subject matter relates generally to multimedia data communications and specifically to multi-axis television information grid navigation.

BACKGROUND

Conventionally, television programming and its metadata guide information has been organized according to original broadcasting schedules into channels, characterized by a given program on a given channel at a given time. End users and broadcasters alike have become accustomed to the ingrained concept of navigating for programs in terms of a two-dimensional program listings grid, upon which for a given time point on a time axis of the grid each channel on a channel axis of the grid has a different program, reflected in language, such as "my favorite show will be on channel six at 8:00 pm." The time/channel information grid, however, is an arbitrary technique for defining a user's desire to find out what is on television.

A television channel represents a specific sequence of programs over time. Broadcasters conventionally determine the programming that they will carry

during a future time interval, making the two axes of channel and time a natural and convenient way to organize television programming content and television programming metadata, that is, the guide information sometimes referred to as electronic program guide (EPG) data.

Fig. 1 shows metadata 100 for a program comprising multimedia content to be shown on television as a "TV show." Of many attributes included in the metadata 100, only two attributes, namely channel 106 and time 108, are utilized to place an identifier of the program, such as the title 102, into a conventional program listings grid (or guide) 104.

Fig. 2 shows the conventional program listings grid 104 of Fig. 1 in greater detail. The channel axis 202 is plotted against the time axis 204, as above, and provides the context for navigating among television programs.

At a given time point 206, usually selected to be the present moment, navigation conventionally consists of traversing from program to program along the channel axis 202. Most conventional TV systems have been designed to navigate according to a channel axis model, in which navigation of both programs and their respective program guide information typically follows the conventional channel axis 202 versus time axis 204 format. That is, as shown in **Fig. 3**, a navigation control, such as a "channel changer" located either on a TV set or on a remote control, traverses the channel axis 202 via, for example, "channel change keys 300," such as a "next" key 302 and a "previous" key 304. An "enter" or "select" key 306 often accompanies channel change keys 300. Some navigation controls that use channel change keys 300 also include left and right keys that may have a "first" and "last" function or perhaps a volume control function. The

illustrated channel change keys 300 are presented, of course, only as one example of conventional navigation controls.

In traversing a channel axis 202, conventional navigation techniques generally hold the time axis 204 constant at a single point representing the present, and cycle through all available channels, displaying the program that is currently playing on each channel in turn, i.e., as a user changes channels the program currently playing on a particular channel automatically displays. The two attributes, channel and time, provide the context for conventional television programming navigation. Navigating from one program or program reference (program ID) to another is sometimes referred to as "scrolling" or "channel-surfing." Each program ID may be used to access stored data in a local, remote, or DVR media store.

Delivery of programs in a sequence over channels is the best-known context for organizing and navigating television content and characterizes a lingering and predominant mindset in the multimedia arts. However, there is now too much multimedia content to rely on conventional navigation techniques. It is awkward to navigate hundreds or even thousands of conventional channels.

Digital storage of television programs with its characteristic random access feature frees navigation systems from accessing only programs that are being broadcast in the present: program access is no longer limited to currently broadcasted content. Accordingly, the amount of content to choose from potentially consists of programs from thousands of channels (several hundred active channels in some present systems) multiplied by the content of each channel over a time period, which can be years. Although the system of navigating programs by a channel axis 202 and a time axis 204 persists, there is far too much

multimedia content to be efficiently and enjoyably navigated by such a system. As Digital Video Recording (DVR) technology evolves, vast stores of content are becoming available to users as "everything on demand," making ease of navigation a priority.

SUMMARY

An exemplary multi-axis television navigation system defines television navigation axes according to attributes of television programs. In one implementation, if a television program has an attribute defining an axis, then the exemplary system links a predefined database query associated with the axis to metadata for the television program. When launched, the predefined query returns a sequence of navigable television programs having the attribute that defines the axis.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a graphic representation of a conventional television program guide information.

Fig. 2 is a graphic representation of a conventional program listings guide.

Fig. 3 is a graphic representation of conventional channel change navigation keys.

Fig. 4 is a graphic representation of an exemplary navigational axis based on an episode attribute.

Fig. 5 is a graphic representation of an exemplary navigational axis based on an actress attribute.

Fig. 6 is a graphic representation of a multiple navigational axes combined in a two-dimensional grid.

Fig. 7 is a graphic representation of various types of navigational axes based on program attributes.

Fig. 8 is a graphic representation of multi-axis television navigation.

Fig. 9 is a block diagram of an exemplary multi-axis television navigation system.

Fig. 10 is a block diagram of an exemplary multi-axis television navigation engine.

Fig. 11 is a block diagram of an exemplary predefined query structure.

Fig. 12 is a block diagram of another exemplary predefined query structure.

Fig. 13 is a graphic representation of an exemplary tree user interface for presenting axes for selection.

Fig. 14 is a graphic representation of an exemplary information page user interface for presenting axes for selection.

Fig. 15 is a flow diagram of an exemplary method of multi-axis television navigation.

DETAILED DESCRIPTION

Overview

The subject matter describes the use of database technology and digital storage to establish new axes, or logical sequences of programs for television exploration ("navigation") in addition to conventional axes known as broadcast channels. When television content is stored digitally, contexts for content discovery and navigation are not limited to conventional chronological sequences, indeed each program with its related attributes can stand on its own, and programs can be combined into navigable sequences based on other contexts besides chronological sequence. Conventional time/channel information grids generated by chronological sequences are an arbitrary technique for defining a user's desire to find out what is on television. The multiple combinable axes described herein allow the user to define custom navigational views that deliver the most useful information suited to the particular user, tailored to the user's taste.

The subject matter enables navigation on arbitrary axes without having to initiate explicit search commands. Predefined axes, such as "actor" and "episode" may be combined with Boolean operators to yield even more navigation contexts.

A navigational axis ("axis") comprises a sequence of programs or more precisely, program identifiers (IDs) that are included in the sequence for some logical reason, usually a common attribute. Conventionally, a channel is one type of axis wherein programs are included in the axis sequence because they all proceed from a common broadcast point. Sequential ordering of programs within an axis may be left random or an ordered sequence may be imposed for an additional logical reason, for example, a conventional channel axis 202 has a particular temporal order that reflects the time that the included programs were

broadcast. **Fig. 4** shows an exemplary axis 400 configured by episode sequence, allowing a user to navigate among programs represented on the episode axis. In this example, the user watching the October 6th episode would be able to navigate to the September 29th episode (e.g., a "previous" key) or the October 13th episode (e.g., a "next" key).

"Navigation," as presented herein with respect to exemplary axes, refers to changing television channels to view different programs. Although with respect to exemplary systems and methods described herein "navigation" can also be applied to browsing just television programming information, i.e., EPG metadata, many implementations of the exemplary systems and methods describe navigating program content, not just program guide information. Hence, for example, when a user selects an exemplary "actor" axis specifying the actor Russell Crowe, navigation controls on the user's equipment change their function from cycling through programs based on broadcast channels to cycling between programs in which Russell Crowe plays a starring role. **Fig. 5** shows another exemplary navigation axis 500 by actor name. In this example, a user can navigate to "Before and After" 502 (e.g., "previous" key) and "Heartburn" 504 (e.g., "next" key) with a single click.

Exemplary methods also allow logical combination of several exemplary axes to achieve even more refined or more extensive navigational contexts, for example, an episode axis specifying "M.A.S.H." and a year axis specifying "1978" result in a different episode of M.A.S.H. from 1978 playing on the user's equipment each time the user actuates "channel change" controls. **Fig. 6** shows another type of exemplary axis combination, in which navigation of two discrete axes is enabled via a two-dimensional grid 600. In this example, analogous to the

time/channel conventional program guide grid 104, a first column 602 represents an axis of different actors, while rows appended to the first column 602 each represent movies of one respective actor. A user can navigate by different actors and by movies of different actors based on the combination of axes.

In one implementation of exemplary navigation, when a user changes programs on an axis, e.g., presses a channel change key to interrupt one episode of GILLIGAN's ISLAND to view another episode of the same show, the interrupted episode resumes where it left off when the user returns. This allows a user to compare movies and episodes and to jump in and out of programs without missing any content.

Practically speaking, in most implementations, an exemplary navigational system might provide only four or five additional predefined axes (that can be further combined with each other) besides the well-known conventional channel and time axes, but many more axes than four or five are possible in an exemplary system since almost unlimited attributes can be used to describe a program.

When a user combines two or more navigational axes to yield a more refined navigational context, for example, axes combined to set up a navigational context of "FRIENDS" episodes AND "Elliot Gould" as actor AND "1997" as the relevant year, the user is relieved of the burden of finding a likeable program amidst large quantities of content. Rather than having to type in three search terms "FRIENDS," "Elliot Gould," and "1997" and supply the Boolean operator "AND" to launch a conventional database search of a content repository, instead links to predetermined queries that compute the axes are supplied in context: links to predefined navigational axes are embedded within overlays, content, and/or metadata being currently displayed in ways that are logically meaningful, i.e., "in

context" as the user navigates. Hence, with a single selection or "click" a user can extend a context attribute from a present program, e.g., "actress Meryl Streep," into an axis having a navigational context of other Meryl Streep programs accessible one after another through the navigation controls that selected the Meryl Streep context attribute. Data associations for fulfilling the predetermined queries can be pre-rendered on a server or in a client. Pre-rendering on a client can offer its own processing advantages.

Exemplary axes are made available to a user as a link, icon, menu tree, etc., for example, in a UI that can be called up to show program metadata, including the links or icons to select axes. Alternatively, axes may be selected using cues, links, controls, etc., that are only on a remote control so that program metadata is never seen while a program associated with the metadata is playing on a display. For example, after pressing a "select" key 306 as a second function key, actuation of the right arrow key might always select an "actor" navigational axis.

In one implementation, axes that are likely to be selected in the context of a TV program that is currently playing are highlighted (e.g., by default) in program data or overlays accessible through the select key 306, etc., that allow axis switching. The user selects among axes and combinations of axes offered, but does not have to create new axes from scratch. In other words, as noted above, axis choices can be laid out for selection in the context of a show that is currently playing and selection of one or more axes launches pre-determined queries.

For comparison, some features of the subject matter render exemplary program navigation on exemplary axes like surfing an Internet web, where hyperlinks are pre-placed in context in displayed content and new content revealed by clicking a hyperlink is pre-rendered. Thus, a user of an exemplary navigation

system can move through content in a finely tuned logical manner using only a TV remote control or other navigation tool that has minimal navigation controls. The subject matter provides the added benefit that each time an exemplary axis and/or context is selected, navigation keys on the user's navigation tool, e.g., a TV remote controller, are converted to "changing channels" according to the newly selected axis and/or context until axes are changed again.

The subject matter takes advantage of the fact that often a user's current program or position within the program guide information provides a great deal of context and can act as a starting point or branching point for further navigation. Thus, if a user is currently focused on a science fiction movie, then an overlay for the science fiction movie providing opportunities to switch axes can include axis selections that represent logical "next steps" in navigation relative to the science fiction movie: for instance, more science fiction movies, more movies by the same director, more movies with the same actor, etc.

This detailed description describes subject matter that includes methods, engines, database structures, and query techniques for creating many types of navigational axes according to attributes of programs and their contexts in addition to conventional channel and broadcast-time attributes.

Multiple Navigational Axes

Fig. 7 shows program metadata 100 for a movie, "The Mountain," available as a program on television. The metadata 100 includes many attributes and respective values for some of the attributes. Some attributes have an "either/or" or digital character (such as, "high definition TV" (HDTV) (y/n?)) while other attributes can assume a variable value within a known set of values, such as an

actress name for an actress attribute, or the year 1978 within a range of years. Attributes may include the type of program (movie, TV series, etc.), program title, alphabetical order of title, year of release, channel, time, first air date, episode order, episode name, genre, actors, writer, director, producer, rating, sound characteristics, video characteristics, language, subtitles, closeness of match to search criteria, popularity, and many other attributes. Programs can have almost unlimited attributes, each of which can be used as a basis for configuring a navigational axis.

When a "director" attribute 702 is available as the context for a "director" navigational axis 704, an instance of the director axis 704 can be configured by also selecting the name "Edward Dmytryk" 706 as a value for the director attribute 702. A sequence of program IDs, if programs for such exist in an operative database, can be computed wherein each program ID points to a program that has the value. The director axis 704 can then be navigated by using navigational controls, such as channel change keys (e.g., 302, 304 and possibly 306), on a user's equipment, such as a TV set, television enabled computer, or TV remote control.

An exemplary application or multi-axis TV navigation engine (e.g., 904 below) changes the functioning of navigation controls (e.g., 300) substituting a new axis for the conventional channel axis 202 usually navigated by the navigation controls (e.g., 300). When a user presses one of the channel change keys 300, for instance, various events can occur depending on the implementation. In one implementation, a television navigation system is concerned mainly (or only) with the program guide information. In this case, navigating a "director"

axis 704 results in the user browsing only the program guide information for each program ID associated with the selected director in the director axis 704.

In another implementation, an exemplary television navigation system displays actual programs for each program ID in an axis. Thus, pressing a channel change key (e.g., 302) changes between programs directed by the person named in the attribute value. If the attribute upon which an axis is based is "episode" then actuating a channel change key 302 would change between episodes of a subject program and display each different episode with each press of a channel change key 302.

Mechanisms for selecting axes to navigate and selecting values for the axes will be discussed more fully below with respect to Figs 13 and 14.

Fig. 8 shows various exemplary navigational axes being navigated in turn by the same navigation controls, such as a set of channel change keys 300. Of course, channel change keys 300 are only one example of navigational controls for multi-axis television navigation. Different operational controls could be used, such as a dial, a touch pad, a keyboard, a mouse, a wheel, trackball, displayed rolling cylinder, onscreen keys, onscreen grid, onscreen cascade of cards, order forms, voice activated navigator, etc. Channel change keys 300, however, demonstrate that exemplary multi-axis navigation can be accomplished via a minimum of controls, such as two or three buttons or keys.

In Fig. 8, a user is currently viewing a program 800, the "Mountain." In one implementation, the user has selected the program 800 by refraining from further channel changes, leaving the program 800 playing on a display. The user may then decide to change navigational axes.

A change of navigational axes may be implemented in various ways. In one implementation an actuation of a "select" key 306 calls up a menu overlaying the program and converts the "previous" and "next" keys 302, 304 to movement keys for navigating the menu until the select key 306 is actuated again, for example, to select a different axis. The menu may provide a list of axes to choose from. In another implementation, a press of a select key 306 allows other keys 302, 304 to cycle choices on a list of axes but a menu is not displayed on the same display as the program, however, the list of axes may be displayed on a remote controller bearing the navigation controls, such as channel change keys 300. In yet another implementation, when only program information (metadata 100) corresponding to a program ID is displayed instead of the program itself, then the metadata 100 may have embedded hyperlinks or icons that allow a user to select axes and/or values for the axes.

Regardless of which implementation of axis selection is employed, the user now changes navigational axes to an "actor" axis 802 that has a "Spencer Tracy" value for the actor attribute and a "movie" axis filter. Some attributes have only one possible value, such as the year of movie release, so that in one application a value for an axis choice defaults to a single value supplied by the metadata 100 without any user intervention. An exemplary application may also default to a first name on a list in the metadata 100, etc. Scrolling while the "Spencer Tracy as actor" axis 802 is selected results in a cycling through available Spencer Tracy movies, each of which are played in turn as the user scrolls. If the user stops on (selects) the movie "The Old Man and the Sea" 804, which starred Spencer Tracy, then "The Old Man and the Sea" 804 becomes a starting point if navigational axes are again switched.

The user switches to a "writer" axis 806. The writer of "The Old Man and the Sea" 804 was Ernest Hemingway, which is inserted as a value in a writer attribute upon which the writer axis 806 is at least in part configured. Now when the user scrolls, available movies based on books written by Ernest Hemingway are played in turn on the television display as the user scrolls with the channel change keys 300. The user pauses on the movie "The Sun Also Rises" 808, which becomes the next starting point for navigation.

The user switches to a "date" axis 810. A value of "1984" is supplied by default as the year of release of the movie "The Sun Also Rises" 808, which was the context or starting point from which axes were changed. As the user scrolls, movies released in 1984 are played in turn on the TV display. The uses stops on or otherwise selects the movie, "Mad Max" 812, which was released in 1984.

The navigational path through television programming content can continue indefinitely with any number of axis changes. A user might continue by reselecting an actor axis 814 and scrolling to arrive at the movie, "Braveheart" 816. From there, a genre axis 818 with the value "war & combat" could be selected and scrolling might arrive at the movie, "Gladiator" 820. Selection of a "director" axis 822 with a value of "Ridley Scott" from "Gladiator" 820 and scrolling might further arrive at the movie, "Blade Runner" 824, and so on.

Exemplary Multi-Axis TV Navigation System

Fig. 9 shows an exemplary multi-axis TV navigation system 900. The illustrated example is only one configuration of an environment suitable for practicing the subject matter, many other configurations are possible.

A server 902, which may serve as a hub for various connected multimedia devices, includes an exemplary multi-axis TV navigation engine ("navigation engine") 904. A headend 906, a node 908, such as a set-top box, a digital video recorder (DVR) 910, a television 912, and a remote controller 914 are also communicatively coupled as illustrated.

The remote controller 914 includes a navigation controller, such as "channel change keys" 300, which provide an agency for selection of program content, program metadata 100, and navigational axes via the multi-axis TV navigation engine 904. Although the navigation engine 904 is shown as included only in the server 902, in alternate implementations, various components of a navigation engine 904 could be located in different parts of the system 900. When the navigation engine 904 changes, computes, or creates an axis, the axis consists of a sequence of program IDs. Program content corresponding to each program ID may be accessed for play from the server 902, the headend 906, the DVR 910, etc. Additionally, if a particular system is set up to display program metadata 100 corresponding to program IDs on an axis, the program metadata 100 may also be accessed from a node 908 in which EPG data, for example, can be stored, or from the navigation engine 904 itself, if it includes a metadata cache.

Exemplary Multi-Axis TV Navigation Engine and Data Structure

Fig. 10 shows an exemplary multi-axis TV navigation engine 904, in greater detail than Fig. 9. An exemplary navigation engine 904 may include a metadata cache 1002, a multi-axis database schema 1004, and a metadata database 1006, which may be broken out into multiple attribute indices 1008. A user interface (UI) 1010 may include an axis selector 1012 that includes a combiner

1014, and a navigation controller 1016, or in some implementations, a navigation controller interface (not shown). The user interface 1010 is coupled to communicate with a query engine 1018 via one or more predefined queries 1020. The query engine is coupled with an axis cache 1022 that includes one or more lists of program identifiers. The axis cache 1022 is communicatively coupled with a content access engine 1026. The illustrated engine can be implemented in hardware and/or software, and is merely one example of a navigation engine 904 suitable for practicing multi-axis TV navigation.

In operation, the metadata cache 1002 receives an input of EPG metadata 100 from a headend 906 of a content provider. The metadata 100 may be in various formats, depending on its source. The multi-axis database schema 1004 arranges, organizes, and/or parses the metadata 100 into a metadata database for compatibility with the query engine 1018 and the predefined queries 1020.

Since many different kinds of databases can be used for the metadata database 1006, the data structure adopted depends on the type. In relational or hierarchical database systems, the multi-axis database schema may index program attributes and/or certain values for the program attributes into index tables or child databases. In an extensible markup language (XML, etc.) system, a multi-axis database schema 1004, such as that represented by the extensible markup language schema definition (XSD) shown in Appendix A, entitled: "Sample Exemplary Schema To Organize EPG Metadata To Support Multi-Axis TV Navigation," specifies a self-referential data structure suitable for multi-axis TV navigation. An XML example of a self-referential metadata database 1004 is shown in Appendix B, entitled "Sample XML Database for Multi-Axis TV Navigation." The sample

metadata database shown in Appendix B includes four television programs and two axes and demonstrates one manner of expressing the definition of an axis.

To recapitulate, a database schema 1004 defines the data representation model: the database schema 1004 organizes metadata 100 in a way that supports declarative queries to generate axes, that is, the sequences of programs or program IDs. From a data representation model, axes may be built, e.g., using indices and predefined queries, on a server 902 and made available for the UI 1010 as iterators. A user can set up a context, e.g., movies with Russell Crowe as an actor, and then iterate or navigate, e.g., "next," "previous," "first," "last," etc. ("first" can show a first episode in a series and "last" shows the last episode), over programs in the constructed axis.

An exemplary database schema 1004 captures program metadata 100 including, e.g., program title information and role information as it is received from a content provider and builds discrete or integrated indices and/or tables representing axes selected to be in the particular exemplary navigational system 900.

In one implementation, a database schema 1004 is in a global listings format (GLF), using a GLF relational model for television programming metadata 100 as described in U.S. Patent Application No. 10/356,694 to Andrew Simms and Samuel Thomas Scott, III, entitled: "Global Listings Format (GLF) for Multimedia Programming Content and Electronic Program Guide (EPG) Information." Television metadata 100 that is received in a GLF (format), which has self-referential structure and contains interlocking and cross-locking lineup, program, and scheduling data is already imbued with a great deal of relational character. GLF data is reliable, because during creation the interlocking and cross-

locking fields are mandatory and may be screened for accuracy. In a GLF schema, moreover, relationships between related pieces of information are built into the fabric of the global listings format.

Whether or not a GLF is used for metadata 100 or the metadata database 1006, it should be noted that selection of a database schema 1004 may depend on the anticipated complexity of an exemplary navigation system 900. In one implementation, each service provider may have different axis index definitions, or in another implementation all axis index definitions are uniform for all participating content providers. For example, a premium service group has more complex content that may be assigned many attributes (e.g., users navigating over HOME BOX OFFICE, the SOPRANOS, R-rated movies, etc.) while a basic service group might be utilizing a different data center with lightweight content, wherein programs are assigned only a few attributes, such as release year and actors' names. Thus, when a subscriber upgrades from a lower service to a premium service more axes may be made available by the content provider.

In one implementation, one or more axis specifications process EPG metadata 100 into indices 1008 set up to assist predefined queries, which construct the axes. In this type of exemplary system, when a user selects an axis in the context of a default or selected attribute, for example an attribute selected from program currently being displayed, a predefined query associated with the selected axis runs through one or more appropriate indices 1008 to construct the selected axis, i.e., the list of program IDs. Depending on the format used for the EPG metadata 100, the formation of workable indices 1008 may require iterating and sorting out the newly revised EPG metadata 100 for each program, one program at a time. Program axis queries against a database 1006 then describe relationships

between the indices 1008, and allow the database 1006 to produce the desired query response, thereby constructing a resulting axis. Thus, a query to the server 902 encapsulates commands to produce content on a new axis. Query composition can span one or more axes. A simple axis represents a single program attribute, whereas a complex axis represents more than one attribute. For example, a simple axis might represents "actor's name" specifying "Smith," while a complex axis might represent "actor's name" specifying "Smith" and "four star rating." Perhaps the user has just purchased a high definition TV and only wants to find HDTV programs amidst the content. If the navigation system 900 allows, the user can add a HDTV axis to specify only HDTV content.

In the exemplary UI 1010, the user may employ the navigation controller 1016 to call up an axis selector 1012, e.g., actuate a select key 306 to produce an onscreen overlay of program metadata 100 having embedded axis links. The navigation controller 1016 may then allow selection of one or more links. In one implementation, each axis selected has a predefined query 1020. Multiple axes may be joined by logical operators via the combiner 1014. For example, Boolean operators, such as AND, OR, XOR, NOT, etc. may be inserted between query terms to produce combination axes. Filtering logic may also be used to hone an axis to a very narrow set of program IDs.

The query engine 1018 may be a database engine created especially for the task of optimized metadata database 1006 queries, but in most implementations, standard database engines and tools, such as those included in MICROSOFT® SQL SERVER 2000 DATABASE SERVER work well in an exemplary navigation engine 904 (Microsoft Corp., Redmond, Wa). The query engine 1008 returns an axis, that is, one or more lists of program identifiers 1024, to an axis

cache 1022. Then, a program content access engine 1026, as directed by the navigation controller 1016, reads a program ID from the list of program identifiers 1024 and retrieves a television program for display. The program content access engine 1016 may be set up to retrieve content from multiple sources, such as a remote headend 906 or storage media, local storage media on a server 902, a networked DVR 910, etc.

Fig. 11 shows an exemplary query structure 1100, illustrating the key-value pair associations with program data. Using a relationship between the key-value pair associations with program data, the query establishes and produces an axis of program data for navigating by an episode attribute of program descriptions.

Fig. 12 shows a query structure 1200, illustrating an actor-based axis construction. Using relationships between the key-value pair associations and the program data, the query establishes an axis of program data for navigating by the actor name attribute of program descriptions.

Exemplary User Interfaces

Exemplary UIs 1010 offer ways to select axes, that is, ways to select among data sets generated by the predefined queries 1020. The selection of multiple axes may be made available in the context of a UI 1010, as noted above, whether the UI 1010 is a grid, a program information page, icons, video thumbnails, etc. For example, in one exemplary UI 1010, clicking on a link or icon for one axis may display a hint, such as a highlight or dotted line outlining a cue word, that an alternate and/or additional axis is available to be iterated over. Or again, clicking on the word "page" on a displayed page of EPG metadata 100 might cause a cue word or icon for a movie axis to be highlighted. Perhaps another click highlights

actors, building a combination of ever more refined navigation contexts using additional axes, with each click.

Fig. 13 shows part of an exemplary program attribute tree 1300 that can be used in various ways as an exemplary UI 1010 or as part of an exemplary UI 1010. In one implementation, the tree is displayed only on a remote controller 914 with navigation controls 1016 for highlighting and selecting attributes corresponding to axes. For many navigation systems 900, the exemplary tree 1300 only needs to include a few member attributes. In another implementation, the exemplary tree 1300 or parts thereof appears onscreen on the same display as the program content, for example, as a menu overlay. There are many types of UIs 1010, however, besides the exemplary tree 1300 that can be used for displaying axes for selection in an exemplary navigation system 900.

Fig. 14 shows an exemplary UI 1010 comprising an onscreen information page 1402 for each program ID on an axis. When a user navigates over the axis, which is associated with a name value for an actress, the respective information page 1402 for each current program ID comes to the front. The actress name (in the solid selection box) 1404 and movie attribute (in the dotted selection box) 1406 indicate that both attributes (name, movie) will be use in a Boolean "AND" fashion to determine navigation behavior. A video window 1408 may be added to each information page to play the television program and/or trailer associated with an information page when it is navigated to and comes to the front.

Fig. 15 is a flow diagram of an exemplary method 1500 of multi-axis TV navigation. This exemplary method 1500 can be performed by a module or engine, such as the exemplary navigation engine 904 shown in Fig. 10. In the flow diagram, the operations are summarized in individual blocks. The operations

may be performed in hardware and/or as machine-readable instructions (software or firmware) that can be executed by a processor.

At block 1502, combinable navigation contexts for creating a list of television programs are selected.

At block 1504, the combinable navigation contexts are logically combined.

At block 1506, a database of television programming metadata is queried for television program identifiers associated with the combined navigation contexts.

At block 1508, television programs associated with the identifiers are presented for navigation.

CONCLUSION

It should be noted that the subject matter described above can be implemented in hardware, in software, or in both hardware and software. In certain implementations, the exemplary system, engine, and related methods may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The subject matter can also be practiced in distributed communications environments where tasks are performed over wireless communication by remote processing devices that are linked through a communications network. In a wireless network, program modules may be located in both local and remote communications device storage media including memory storage devices.

The foregoing discussion describes exemplary systems and methods for multi-axis television navigation. Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as exemplary forms of implementing the claims.